Specialising Linear Algebra for Geometry

Guy Davidson

C++ On Sea 15/07/2020

@hatcat01

#include <C++>

https://www.includecpp.org/

What to expect...

What to expect...

0. A brief history of geometry

1. Linear algebra and geometry

2. Lines and curves

3. Polygons, regular and irregular

4. Intersection and precision

5. Summary of classes and functions

A Brief History Of Geometry

“The branch of mathematics concerned with questions of shape, size, relative position of figures and the properties of space.”

A Brief History Of Geometry

“The branch of mathematics concerned with questions of shape, size, relative position of figures and the properties of space.”

A Brief History Of Geometry

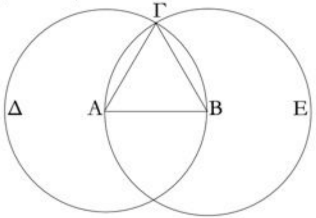
“The branch of mathematics concerned with questions of shape, size, relative position of figures and the properties of space.”

A Brief History Of Geometry

A Brief History Of Geometry

αʹ.

Ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τρίγωνον ἰσόπλευρον συστήσασθαι.



Ἔστω ἡ δοθεῖσα εὐθεῖα πεπερασμένη ἡ ΑΒ.

Δεῖ δὴ ἐπὶ τῆς ΑΒ εὐθείας τρίγωνον ἰσόπλευρον συστήσασθαι.

Κέντρῳ μὲν τῷ Α διαστήματι δὲ τῷ ΑΒ κύκλος γεγράφθω ὁ ΒΓΔ, καὶ πάλιν κέντρῳ μὲν τῷ Β διαστήματι δὲ τῷ ΒΑ κύκλος γεγράφθω ὁ ΑΓΕ, καὶ ἀπὸ τοῦ Γ σημείου, καθ' ὃ τέμνουσιν ἀλλήλους οἱ κύκλοι, ἐπί τὰ Α, Β σημεῖα ἐπεζεύχθωσαν εὐθεῖαι αἱ ΓΑ, ΓΒ.

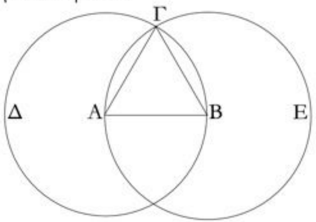
Καὶ ἐπεὶ τὸ Α σημεῖον κέντρον ἐστὶ τοῦ ΓΔΒ κύκλου, ἴση ἐστὶν ἡ ΑΓ τῇ ΑΒ· πάλιν, ἐπεὶ τὸ Β σημεῖον κέντρον ἐστὶ τοῦ ΓΑΕ κύκλου, ἴση ἐστὶν ἡ ΒΓ τῇ ΒΑ. ἐδείχθη δὲ καὶ ἡ ΓΑ τῇ ΑΒ ἴση· ἑκατέρα ἄρα τῶν ΓΑ, ΓΒ τῇ ΑΒ ἐστιν ἴση. τὰ δὲ τῷ αὐτῷ ἴσα καὶ ἀλλήλοις ἐστὶν ἴσα· καὶ ἡ ΓΑ ἄρα τῇ ΓΒ ἐστιν ἴση· αἱ τρεῖς ἄρα αἱ ΓΑ, ΑΒ, ΒΓ ἴσαι ἀλλήλαις εἰσίν.

Ἰσόπλευρον ἄρα ἐστὶ τὸ ΑΒΓ τρίγωνον. καὶ συνέσταται ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τῆς ΑΒ. ὅπερ ἔδει ποιῆσαι.

A Brief History Of Geometry

α′.

Ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τρίγωνον ἰσόπλευρον συστήσασθαι.



Ἔστω ἡ δοθεῖσα εὐθεῖα πεπερασμένη ἡ ΑΒ.

Δεῖ δὴ ἐπὶ τῆς ΑΒ εὐθείας τρίγωνον ἰσόπλευρον συστήσασθαι.

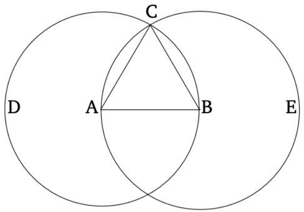
Κέντρῳ μὲν τῷ Α διαστήματι δὲ τῷ ΑΒ κύκλος γεγράφθω ὁ ΒΓΔ, καὶ πάλιν κέντρῳ μὲν τῷ Β διαστήματι δὲ τῷ ΒΑ κύκλος γεγράφθω ὁ ΑΓΕ, καὶ ἀπὸ τοῦ Γ σημείου, καθ' ὃ τέμνουσιν ἀλλήλους οἱ κύκλοι, ἐπί τὰ Α, Β σημεῖα ἐπεζεύχθωσαν εὐθεῖαι αἱ ΓΑ, ΓΒ.

Καὶ ἐπεὶ τὸ Α σημεῖον κέντρον ἐστὶ τοῦ ΓΔΒ κύκλου, ἴση ἐστὶν ἡ ΑΓ τῇ ΑΒ· πάλιν, ἐπεὶ τὸ Β σημεῖον κέντρον ἐστὶ τοῦ ΓΑΕ κύκλου, ἴση ἐστὶν ἡ ΒΓ τῇ ΒΑ. ἐδείχθη δὲ καὶ ἡ ΓΑ τῇ ΑΒ ἴση· ἑκατέρα ἄρα τῶν ΓΑ, ΓΒ τῇ ΑΒ ἐστιν ἴση. τὰ δὲ τῷ αὐτῷ ἴσα καὶ ἀλλήλοις ἐστὶν ἴσα· καὶ ἡ ΓΑ ἄρα τῇ ΓΒ ἐστιν ἴση· αἱ τρεῖς ἄρα αἱ ΓΑ, ΑΒ, ΒΓ ἴσαι ἀλλήλαις εἰσίν.

Ἰσόπλευρον ἄρα ἐστὶ τὸ ΑΒΓ τρίγωνον. καὶ συνέσταται ἐπὶ τῆς δοθείσης εὐθείας πεπερασμένης τῆς ΑΒ. ὅπερ ἔδει ποιῆσαι.

Proposition 1

To construct an equilateral triangle on a given finite straight-line.



Let AB be the given finite straight-line.

So it is required to construct an equilateral triangle on the straight-line AB.

Let the circle BCD with center A and radius AB have been drawn [Post. 3], and again let the circle ACE with center B and radius BA have been drawn [Post. 3]. And let the straight-lines CA and CB have been joined from the point C, where the circles cut one another,† to the points A and B (respectively) [Post. 1].

And since the point A is the center of the circle CDB, AC is equal to AB [Def. 1.15]. Again, since the point B is the center of the circle CAE, BC is equal to BA [Def. 1.15]. But CA was also shown (to be) equal to AB. Thus, CA and CB are each equal to AB. But things equal to the same thing are also equal to one another [C.N. 1]. Thus, CA is also equal to CB. Thus, the three (straight-lines) CA, AB, and BC are equal to one another.

Thus, the triangle ABC is equilateral, and has been constructed on the given finite straight-line AB. (Which is) the very thing it was required to do.

@hatcat01

A Brief History Of Geometry

René Descartes

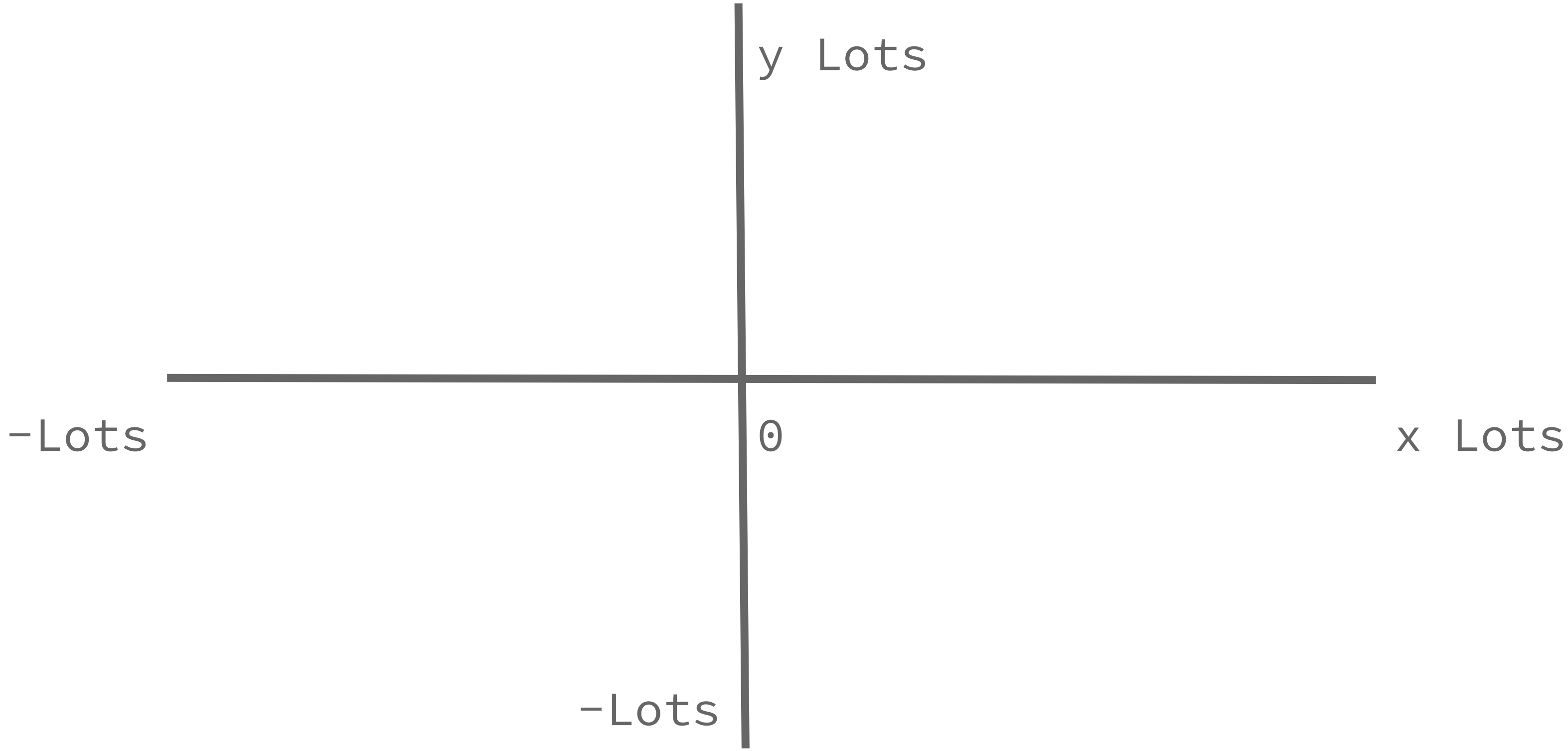
b. 31st March 1596

d. 11th February 1650

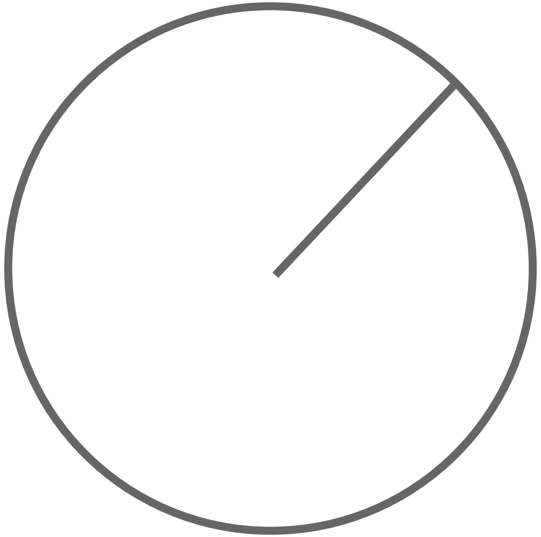
A Brief History Of Geometry

-Lots 0 Lots

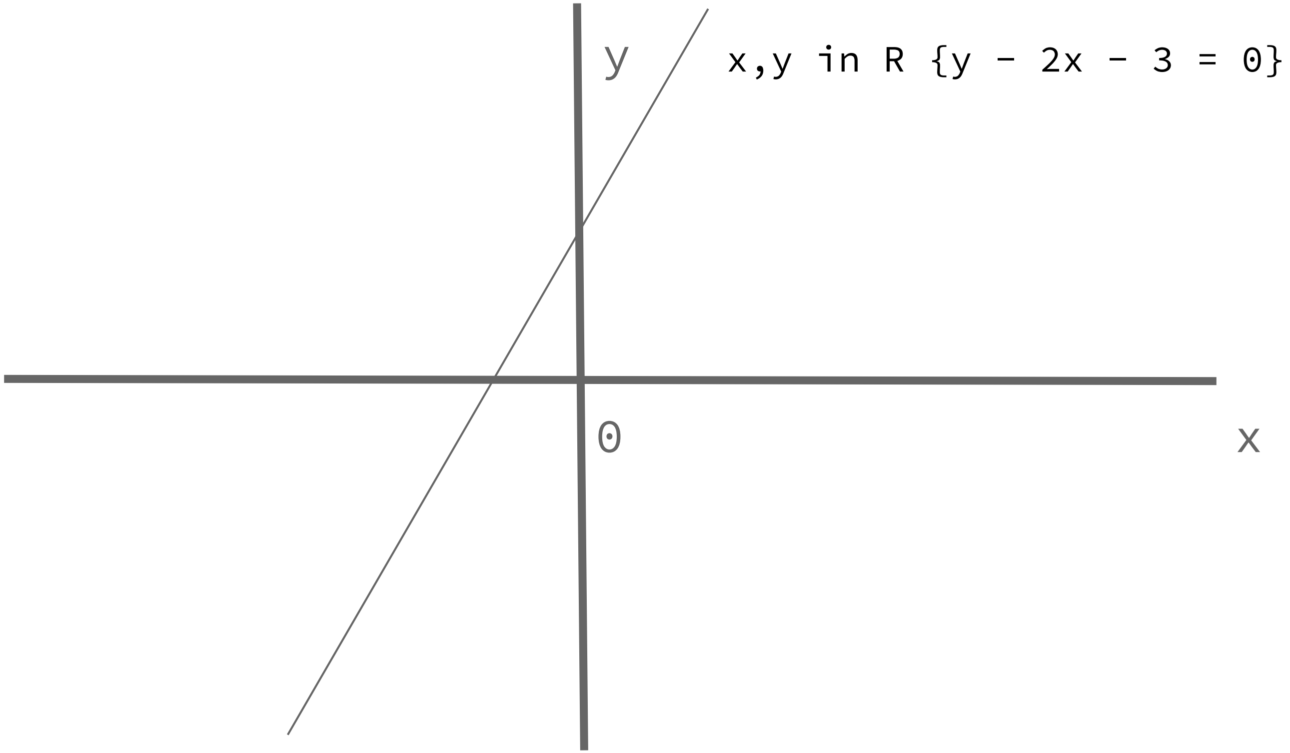
A Brief History Of Geometry



A Brief History Of Geometry

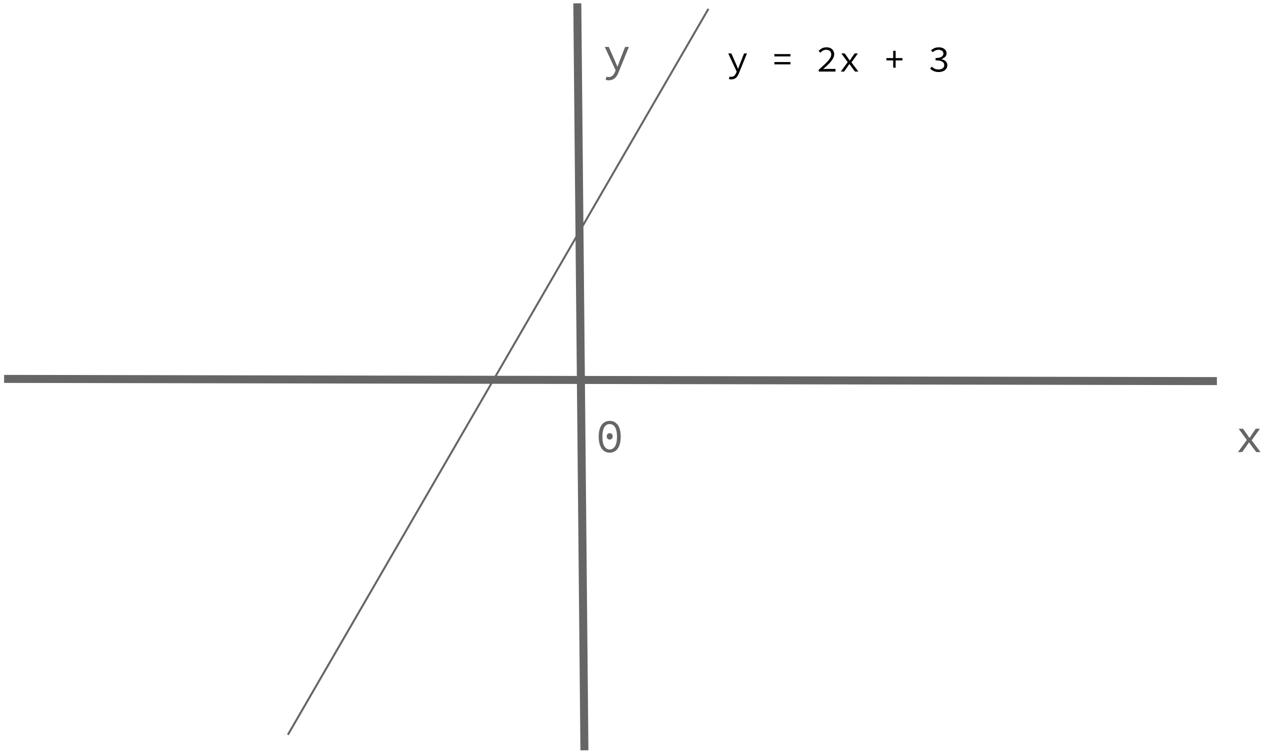


A Brief History Of Geometry

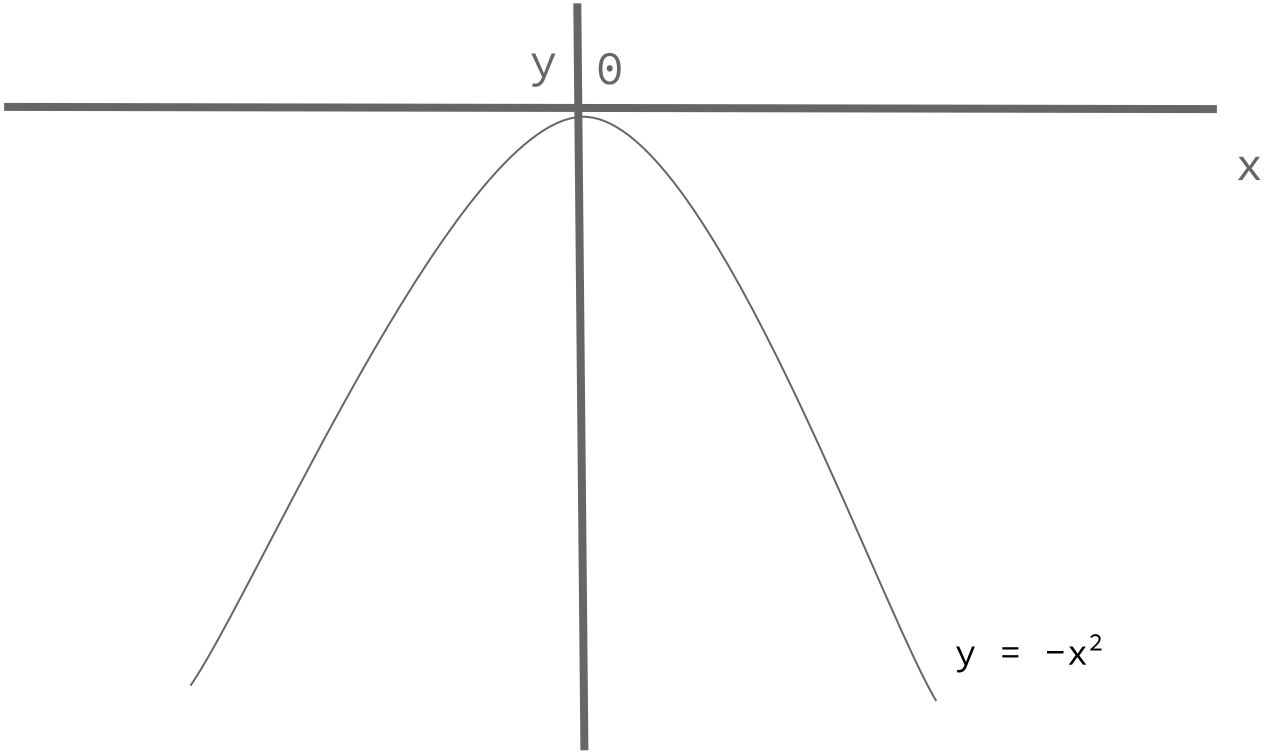


@hatcat01

A Brief History Of Geometry

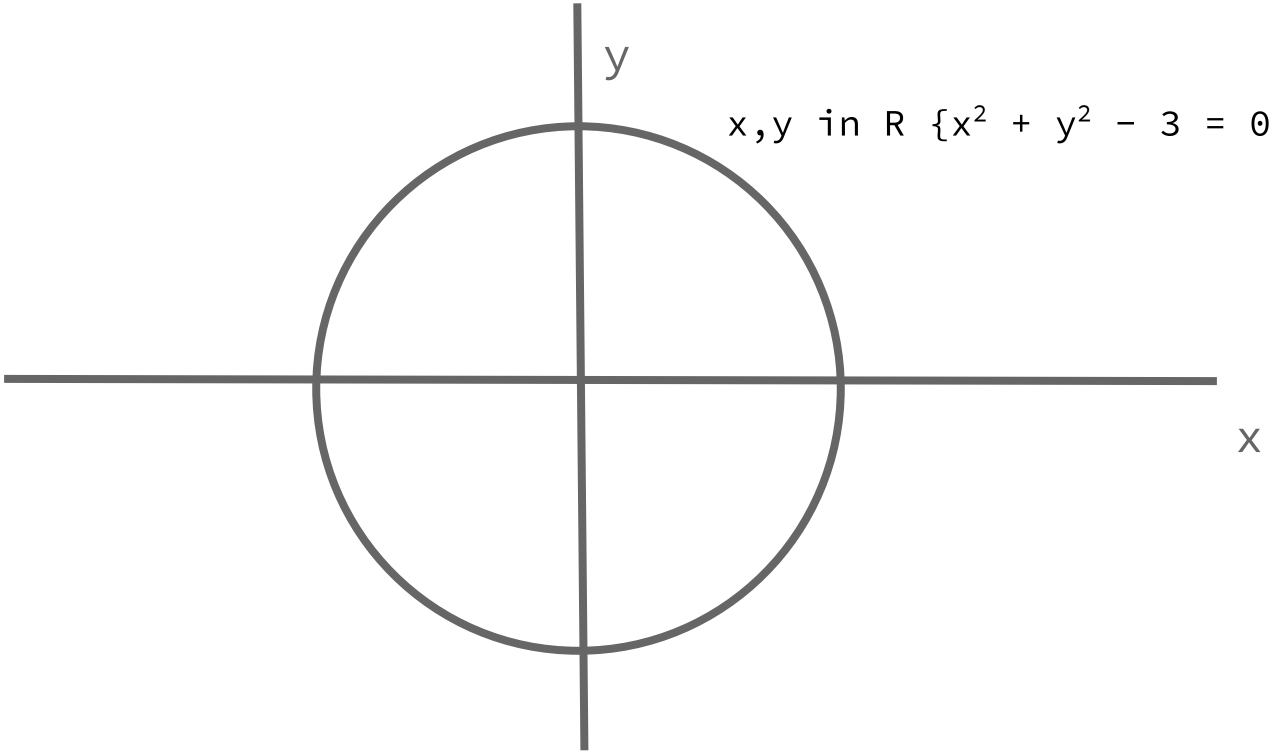


A Brief History Of Geometry

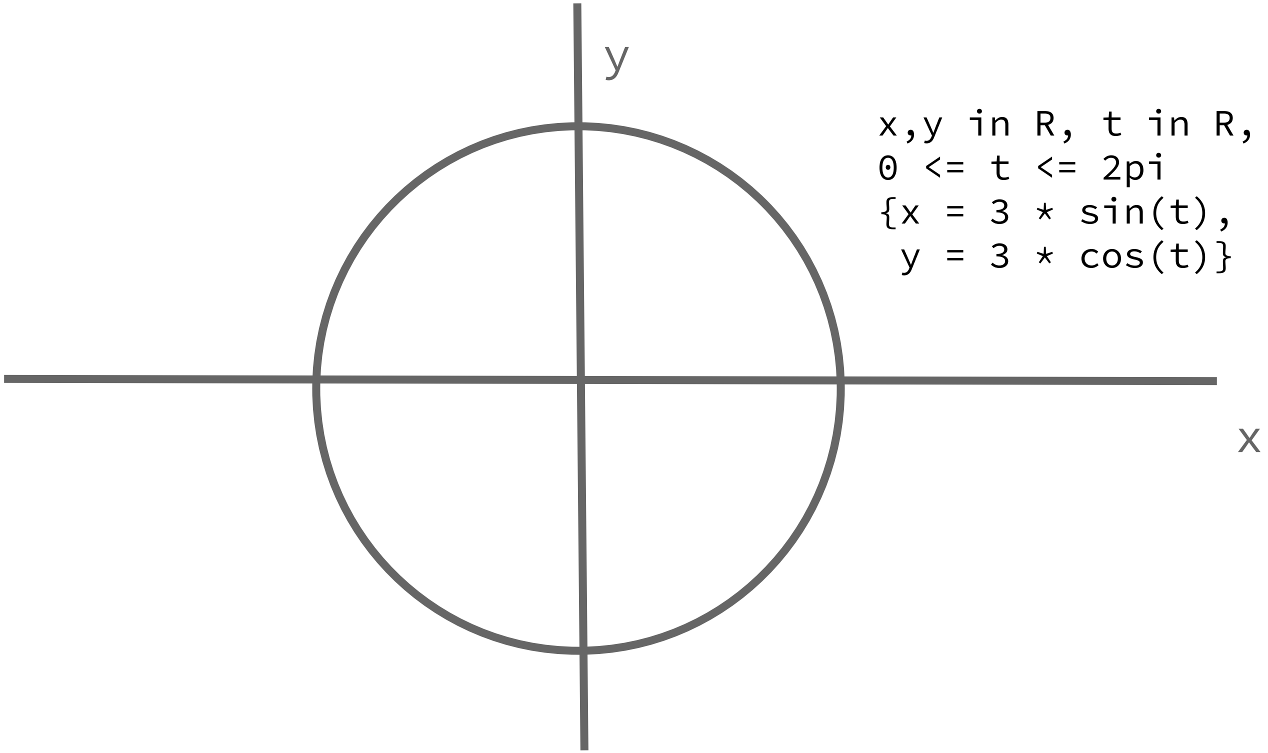


A Brief History Of Geometry

x,y in R $\{x^2 + y^2 - 3 = 0\}$



A Brief History Of Geometry




```
x,y in R, t in R,
0 <= t <= 2pi
{x = 3 * sin(t),
 y = 3 * cos(t)}
```

A Brief History Of Geometry

Other geometries

A Brief History Of Geometry

Other geometries:

Differential geometry

A Brief History Of Geometry

Other geometries:

Differential geometry

Algebraic geometry

A Brief History Of Geometry

Other geometries:

Differential geometry

Algebraic geometry

Cartesian geometry

What to expect...

0. A brief history of geometry
1. **Linear algebra and geometry**
2. Lines and curves
3. Polygons, regular and irregular
4. Intersection and precision
5. Summary of classes and functions

Linear Algebra And Geometry

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b$$

Linear Algebra And Geometry

$$a_1x_1 + a_2x_2 + \dots + a_nx_n = b$$

$$a_1x_1 + a_2x_2 = b$$

@hatcat01

Linear Algebra And Geometry

$a_1x_1 + a_2x_2 + … + a_nx_n = b$

$a_1x_1 + a_2x_2 = b$

$ax + by = c$

Linear Algebra And Geometry

$a_1x_1 + a_2x_2 + … + a_nx_n = b$

$a_1x_1 + a_2x_2 = b$

$ax + by = c$

$by = -ax + c$

Linear Algebra And Geometry

$a_1x_1 + a_2x_2 + … + a_nx_n = b$

$a_1x_1 + a_2x_2 = b$

$ax + by = c$

$by = -ax + c$

$y = mx + c$

@hatcat01

Linear Algebra And Geometry

(x, y)

Linear Algebra And Geometry

(x, y)

Translate

(x, y) + (a, b) = (x+a, y+b)

Linear Algebra And Geometry

Scale

```
(x, y) * 2 = (2x, 2y)

(x, y) * (2 0) = (2x, 2y)
         (0 2)
```

Linear Algebra And Geometry

Shear

```
(x, y) * (1 2) = (x, 2x + y)
         (0 1)
```

Linear Algebra And Geometry

Reflect

```
(x, y) * (1  0) = (x, -y)
         (0 -1)
```

Linear Algebra And Geometry

Rotate

```
(x, y) * (cos a -sin a)
         (sin a  cos a)

= (x*cos a + y*sin a,
         -x*sin a + y*cos a)
```

@hatcat01

Linear Algebra And Geometry

Boost.Geometry

Linear Algebra And Geometry

Boost.Geometry

Barend Gehrels

Linear Algebra And Geometry

Boost.Geometry

Barend Gehrels

Geometry classes

Linear Algebra And Geometry

Boost.Geometry

Barend Gehrels

Geometry classes

Dimension agnostic

Linear Algebra And Geometry

Boost.Geometry

Barend Gehrels

Geometry classes

Dimension agnostic

Distance

Linear Algebra And Geometry

Boost.Geometry

Barend Gehrels

Geometry classes

Dimension agnostic

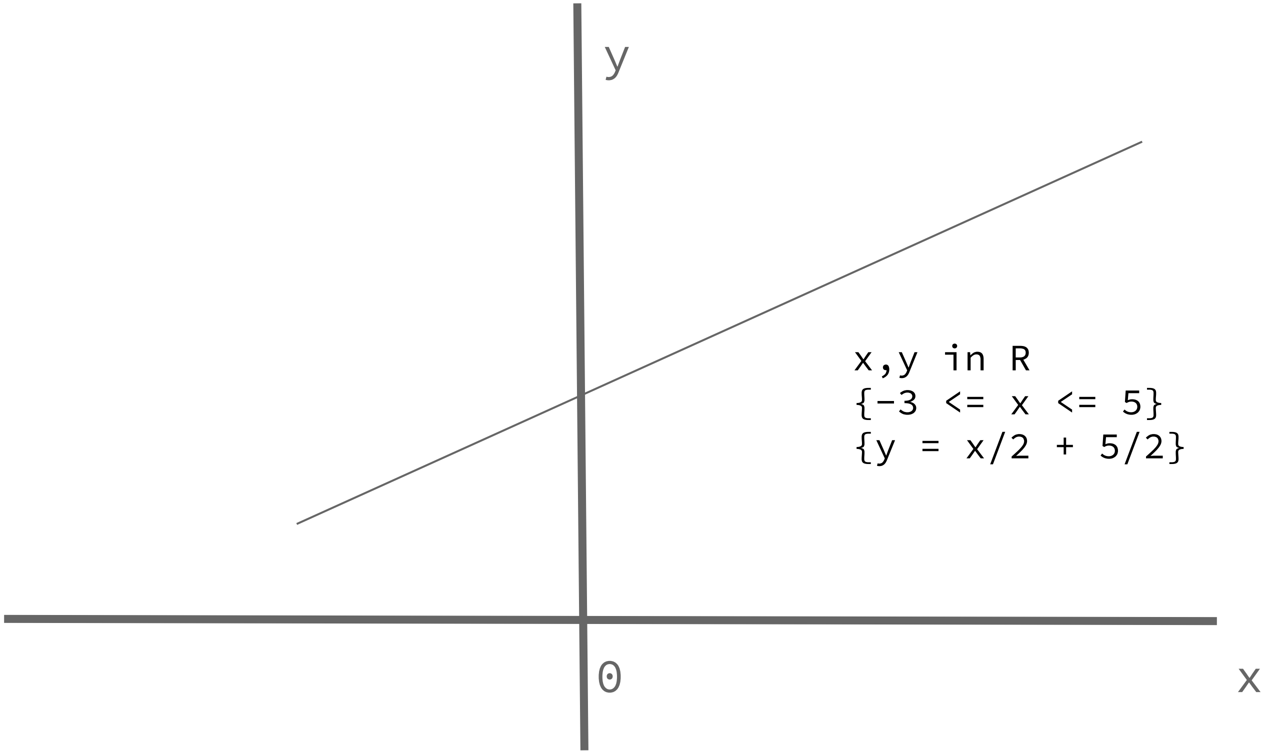
Distance

Coordinate-system agnostic

What to expect...

0. A brief history of geometry
1. Linear algebra and geometry
2. **Lines and curves**
3. Polygons, regular and irregular
4. Intersection and precision
5. Summary of classes and functions

Lines And Curves




```
x,y in R
{-3 <= x <= 5}
{y = x/2 + 5/2}
```

Lines And Curves

```
struct line
{
  float gradient;
  float y_intercept;
};
```

Lines And Curves

```
struct line
{
  float gradient;
  float y_intercept;
};

struct line_segment
{
  point p1;
  point p2;
};
```

Lines And Curves

Q

Lines And Curves

Q

3244.7482

Lines And Curves

$$b_{\nu,n}(x) = \binom{n}{\nu} x^{\nu} (1-x)^{n-\nu}, \quad \nu = 0, \ldots, n,$$

Lines And Curves

$$b_{2,5}(x) = \binom{5}{2} x^2 (1 - x)^3 = 10x^2 (1 - x)^3$$

Lines And Curves

$$B_n(x) = \sum_{\nu=0}^{n} \beta_\nu b_{\nu,n}(x)$$

Lines And Curves

```
class line
{
  std::vector<point> points;
};
```

Lines And Curves

```
class line
{
  float gradient;
  float intercept;
};
```

Lines And Curves

```
class line
{
  float gradient;
  float intercept;
  point p1;
  point p2;
};
```

Lines And Curves

```
class line
{
  float gradient;
  float intercept;
  point p_begin;
  point p_end;
};
```

Lines And Curves

Curves

Lines And Curves

y

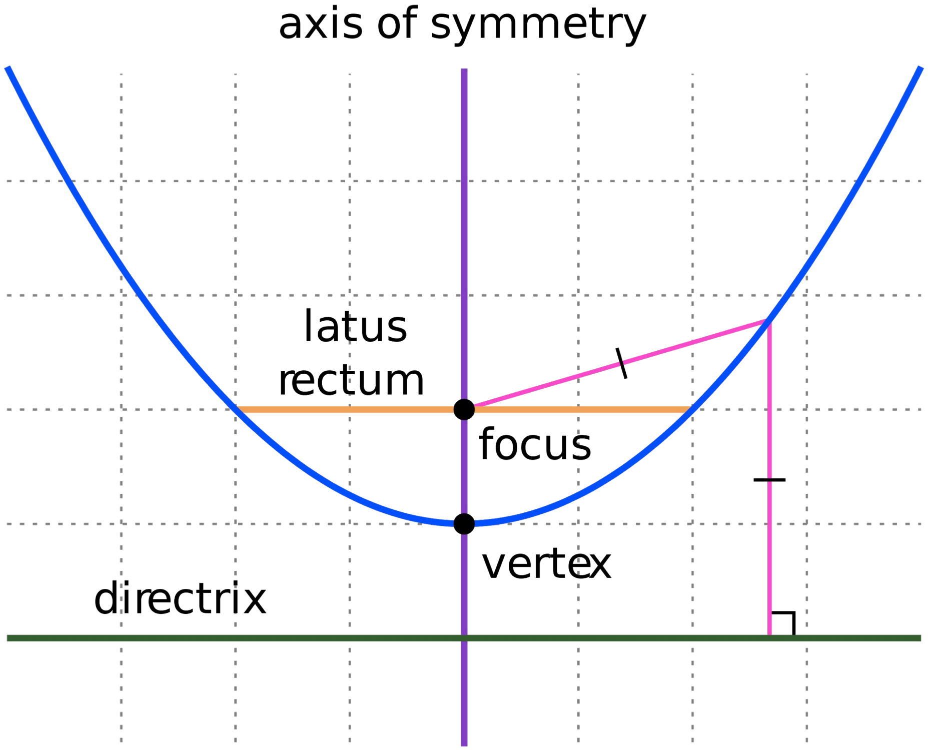
$y = x^2$

0

x

@hatcat01

Lines And Curves



Lines And Curves

```
class curve
{
  point p1;
  point p2;
  point c1;
  point c2;
};
```

Lines And Curves

```
class curve
{
  point p1;
  point p2;
  point c1;
  point c2;
};
```

Quadratic Bézier curve

@hatcat01

Lines And Curves

$$\mathbf{B}(t) = \sum_{i=0}^{n} b_{i,n}(t)\mathbf{P}_i, \quad 0 \leq t \leq 1$$

Lines And Curves

$$\mathbf{B}(t) = \mathbf{P}_0 + t(\mathbf{P}_1 - \mathbf{P}_0) = (1 - t)\mathbf{P}_0 + t\mathbf{P}_1 \, , 0 \leq t \leq 1$$

Lines And Curves

$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]\ , 0 \leq t \leq 1,$$

Lines And Curves

$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]\ , 0 \le t \le 1,$$

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2\ , 0 \le t \le 1.$$

Lines And Curves

$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]\ , 0 \le t \le 1,$$

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2\ , 0 \le t \le 1.$$

$$\mathbf{B}(t) = \mathbf{P}_1 + (1-t)^2(\mathbf{P}_0 - \mathbf{P}_1) + t^2(\mathbf{P}_2 - \mathbf{P}_1)\ , 0 \le t \le 1$$

Lines And Curves

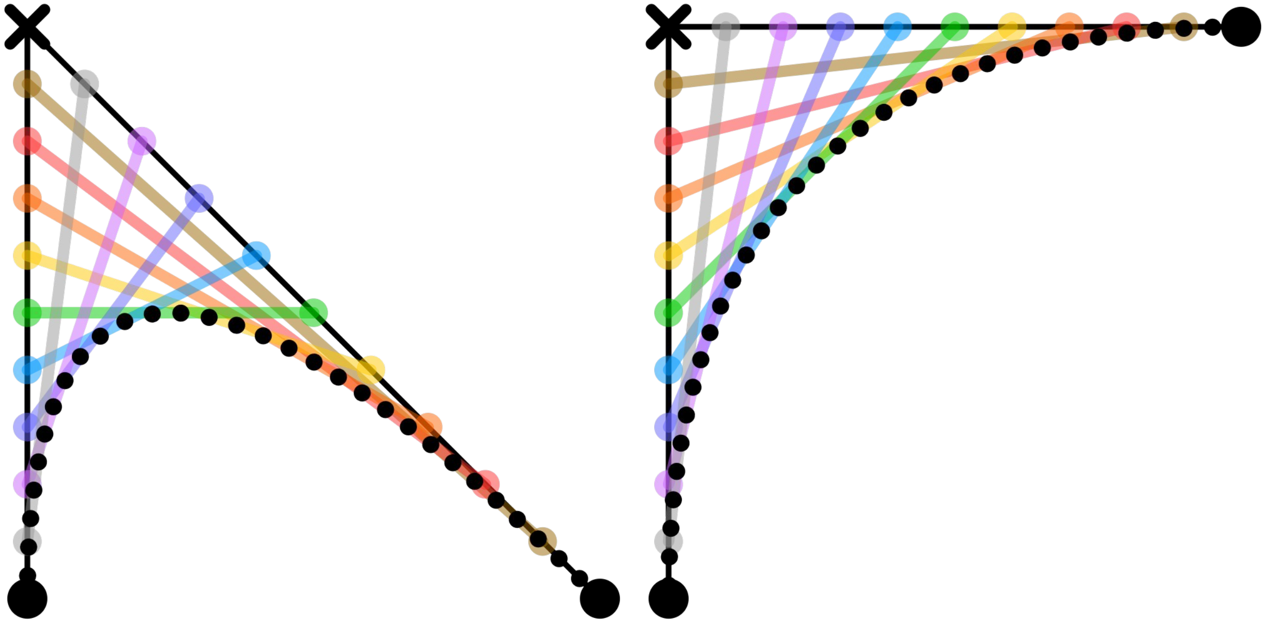
$$\mathbf{B}(t) = (1-t)[(1-t)\mathbf{P}_0 + t\mathbf{P}_1] + t[(1-t)\mathbf{P}_1 + t\mathbf{P}_2]\ , 0 \le t \le 1,$$

$$\mathbf{B}(t) = (1-t)^2\mathbf{P}_0 + 2(1-t)t\mathbf{P}_1 + t^2\mathbf{P}_2\ , 0 \le t \le 1.$$

$$\mathbf{B}(t) = \mathbf{P}_1 + (1-t)^2(\mathbf{P}_0 - \mathbf{P}_1) + t^2(\mathbf{P}_2 - \mathbf{P}_1)\ , 0 \le t \le 1$$

$$\mathbf{B}'(t) = 2(1-t)(\mathbf{P}_1 - \mathbf{P}_0) + 2t(\mathbf{P}_2 - \mathbf{P}_1)$$

Lines And Curves



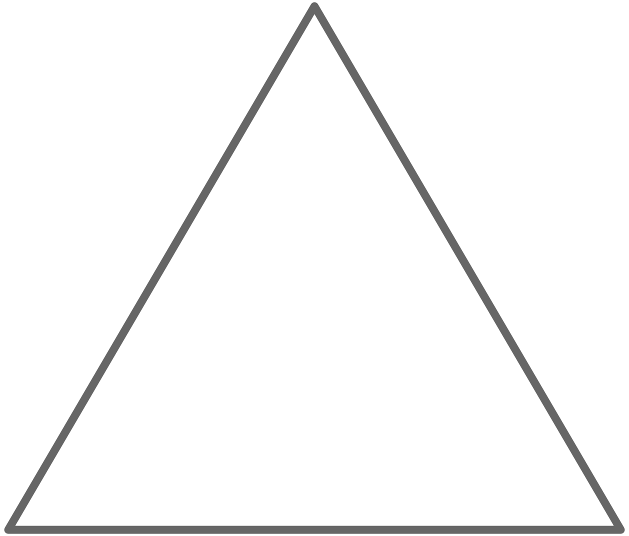
Lines And Curves

```
class curve
{
  point p1;
  point p2;
  point control_point;
};
```

What to expect...

0. A brief history of geometry
1. Linear algebra and geometry
2. Lines and curves
3. **Polygons, regular and irregular**
4. Intersection and precision
5. Summary of classes and functions

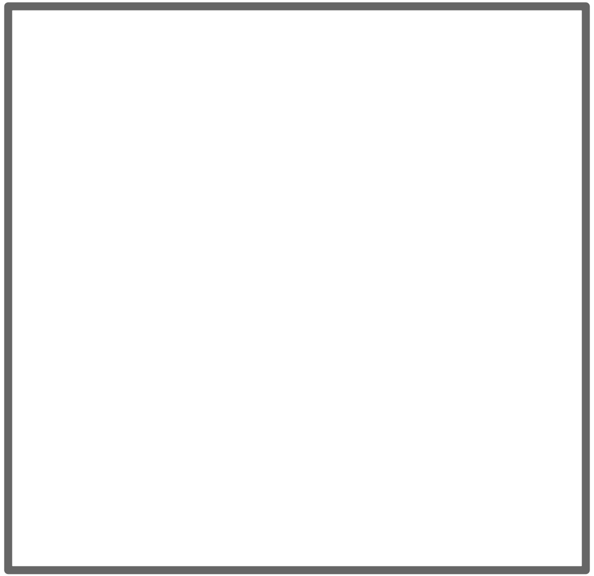
Polygons, Regular And Irregular



Polygons, Regular And Irregular

```
class triangle
{
  point p1;
  point p2;
  point p3;
};
```

Polygons, Regular And Irregular



Polygons, Regular And Irregular

```
class square
{
  point p1;
  point p2;
  point p3;
  point p4;
};
```

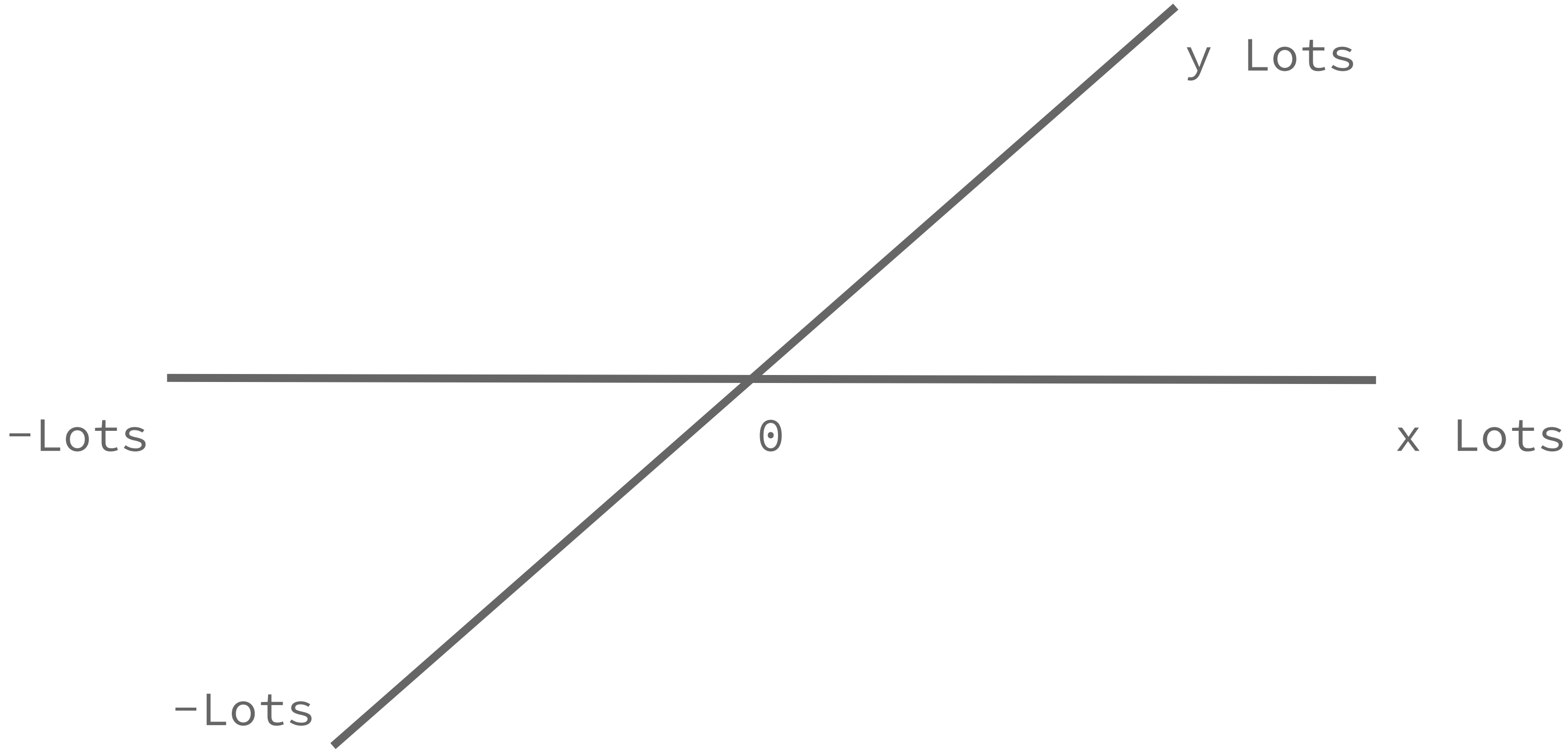
Polygons, Regular And Irregular

```
class polygon
{
  std::vector<point>;
};
```

Polygons, Regular And Irregular

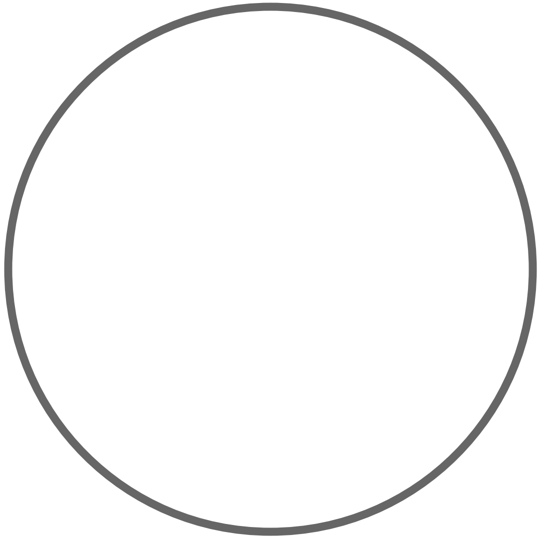
```
class regular_polygon
{
  point centre;
  point p;
  size_t vertex_count;
  float orientation;
};
```

Polygons, Regular And Irregular



@hatcat01

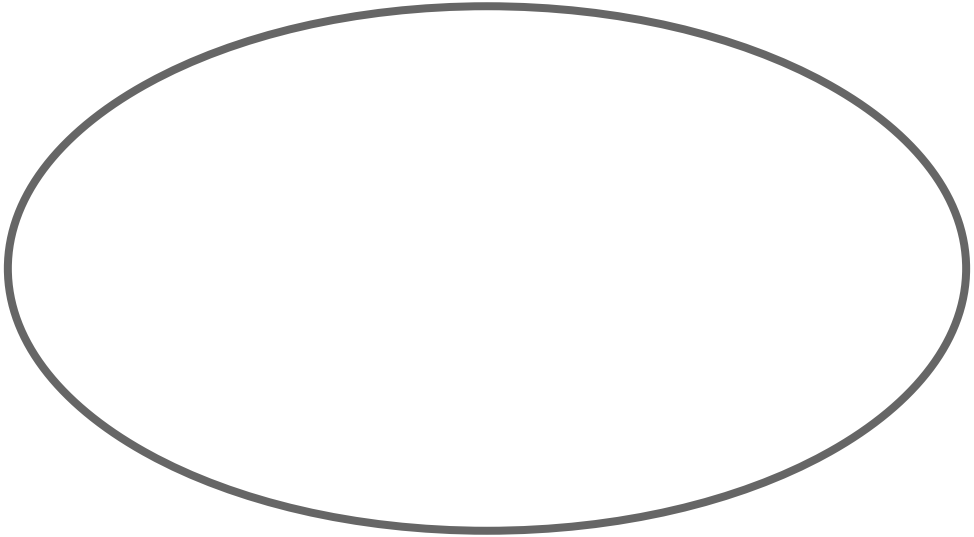
Polygons, Regular And Irregular



Polygons, Regular And Irregular

```
class circle
{
  point centre;
  float radius;
};
```

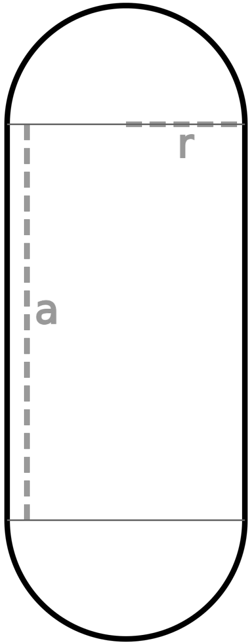
Polygons, Regular And Irregular



Polygons, Regular And Irregular

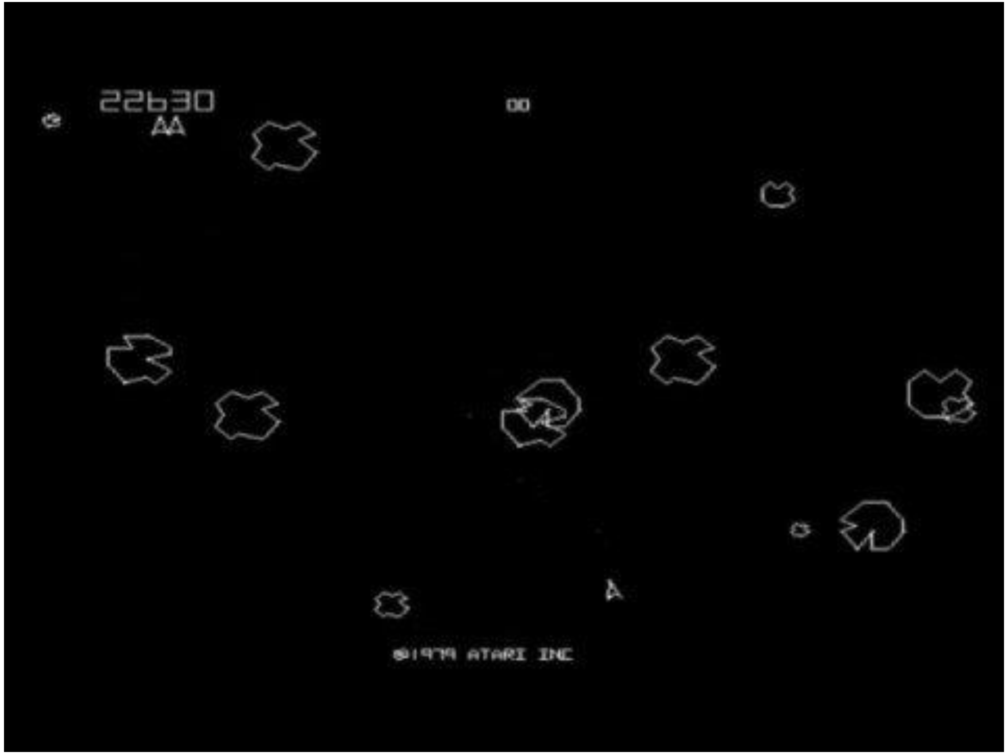
```
class ellipse
{
  point focus_1;
  point focus_2;
  float radius;
  bool major;
};
```

Polygons, Regular And Irregular



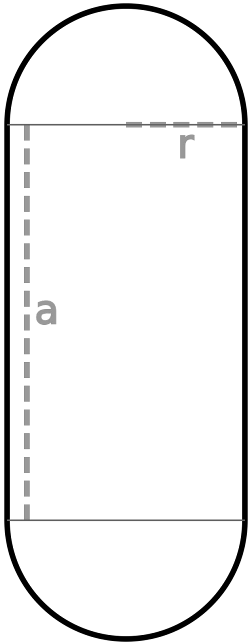
@hatcat01

Polygons, Regular And Irregular

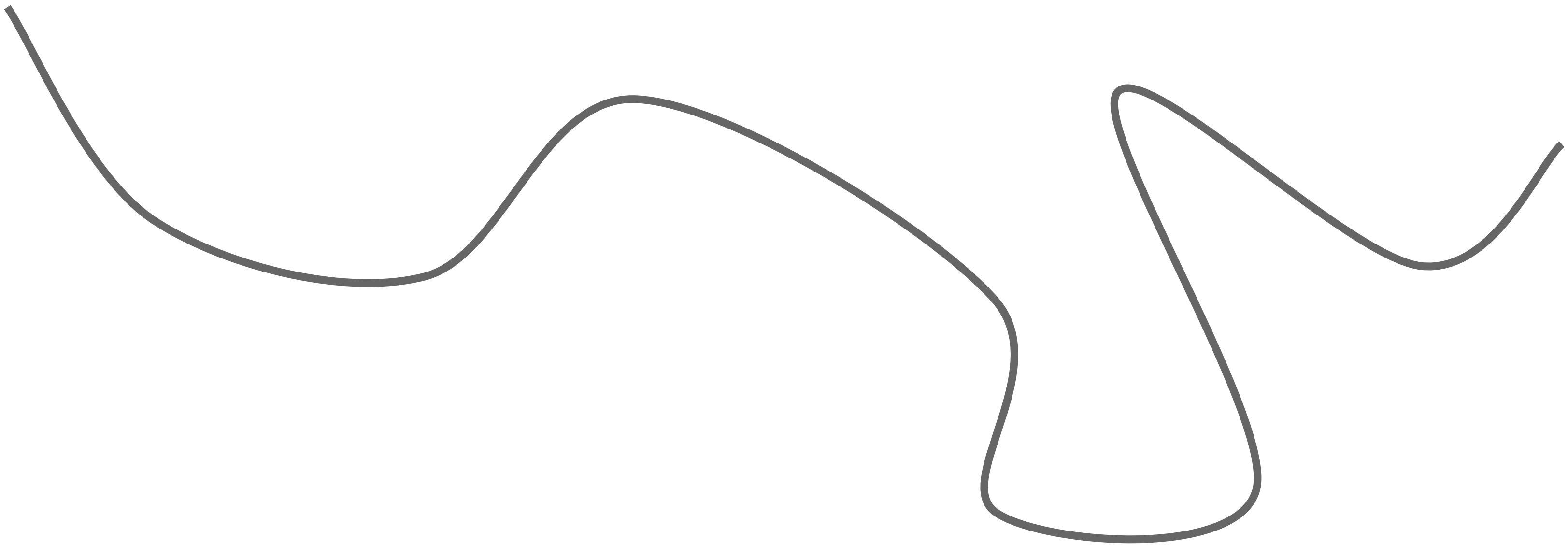


Polygons, Regular And Irregular

```
class stadium
{
  point focus_1;
  point focus_2;
  float radius;
};
```



Polygons, Regular And Irregular



Polygons, Regular And Irregular

```
class polycurve
{
  std::vector<std::curve> segments;
};
```

Polygons, Regular And Irregular

```
class polycurve
{
  std::vector<std::pair<std::point>> segments;
  std::optional<point> end_point;
};
```

Polygons, Regular And Irregular

$$\mathbf{p}(u,v) = \sum_{i=0}^{n} \sum_{j=0}^{m} B_i^n(u)\ B_j^m(v)\ \mathbf{k}_{i,j}$$

@hatcat01

What to expect...

0. A brief history of geometry
1. Linear algebra and geometry
2. Lines and curves
3. Polygons, regular and irregular
4. **Intersection and precision**
5. Summary of classes and functions

Intersection And Precision

Intersection

Intersection And Precision

Intersection

```
y = x - 1
y = 2x - 4
```

Intersection And Precision

Intersection

```
y = x - 1
y = 2x - 4

0 = x - 3
x = 3
```

Intersection And Precision

Intersection

$y = x^2$

$y = x + 3.9$

Intersection And Precision

Intersection

$y = x^2$

$y = x + 3.9$

$0 = x^2 - x - 3.9$

$x = 0.5 \pm \sqrt{}(4.15)$

Intersection And Precision

Intersection

```
y = x - 2.3
y = x/3
```

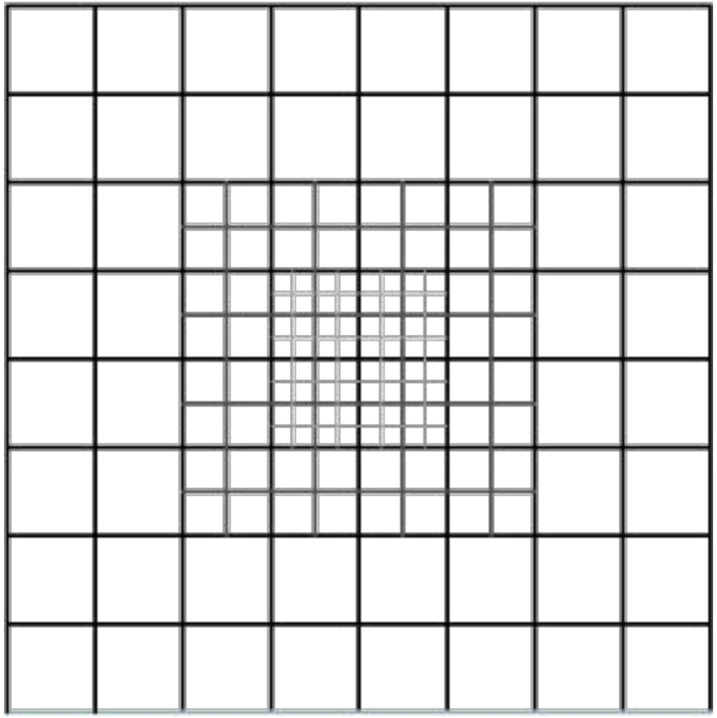
Intersection And Precision

Intersection

```
y = x - 2.3
y = x/3

0 = 2x/3 - 2.3
x = 3.45
```

Intersection And Precision



Intersection And Precision

A blow to the head

Intersection And Precision

A blow to the head

Swords

Intersection And Precision

A blow to the head

Swords

Fast and thin

Intersection And Precision

A blow to the head

Swords

Fast and thin

```
bool intersects(line a, line b);
```

Intersection And Precision

A blow to the head

Swords

Fast and thin

`bool intersects(line a, line b);`

`FLT_MIN vs FLT_EPSILON`

Intersection And Precision

```
bool intersects(line a, line b, float epsilon);
```

What to expect

0. A brief history of geometry
1. Linear algebra and geometry
2. Lines and curves
3. Polygons, regular and irregular
4. Intersection and precision
5. **Summary of classes and functions**

Summary Of Classes And Functions

`std::math`

Summary Of Classes And Functions

```
std::math

using float_2 =
  std::math::vector<fs_vector_engine<float, 2>>;

using float_22 =
  std::math::matrix<fs_matrix_engine<float, 2, 2>>;
```

Summary Of Classes And Functions

```
std::math

using float_2 =
  std::math::vector<fs_vector_engine<float, 2>>;

using float_22 =
  std::math::matrix<fs_matrix_engine<float, 2, 2>>;

Implementer specialisation
```

Summary Of Classes And Functions

`std::math::path`

Summary Of Classes And Functions

`std::math::path`

3 (or 4?) control points

Summary Of Classes And Functions

std::math::path

3 (or 4?) control points

std::math::polyline

Summary Of Classes And Functions

`std::math::path`

3 (or 4?) control points

`std::math::polyline`

`std::math::polycurve`

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;
```

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;

template <typename coordinate_system>
using triangle = regular_polygon<3, coordinate_system>;
```

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;

template <typename coordinate_system>
using triangle = regular_polygon<3, coordinate_system>;

class circle;
```

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;

template <typename coordinate_system>
using triangle = regular_polygon<3, coordinate_system>;

class circle;

class ellipse;
```

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;

template <typename coordinate_system>
using triangle = regular_polygon<3, coordinate_system>;

class circle;

class ellipse;

class stadium; (?)
```

Summary Of Classes And Functions

```
template <int N, typename coordinate_system>
class regular_polygon;

template <typename coordinate_system>
using triangle = regular_polygon<3, coordinate_system>;

class circle;

class ellipse;

class stadium; (?)

class patch;
```

Summary Of Classes And Functions

`intersect()`

Summary Of Classes And Functions

`intersect()`

`distance()`

Summary Of Classes And Functions

`intersect()`

`distance()`

`length()`

Summary Of Classes And Functions

`contains()`

Summary Of Classes And Functions

`contains()`

`area()`

Summary Of Classes And Functions

`contains()`

`area()`

`perimeter()`

Summary Of Classes And Functions

`contains()`

`area()`

`perimeter()`

`centroid()`

Summary Of Classes And Functions

`contains()`

`area()`

`perimeter()`

`centroid()`

`envelope()`

THANK YOU!

Ask me two questions.

Credits and acknowledgements

This was built using **show.cpp** which you can find at github.com/hatcat/show.cpp along with this presentation.

show.cpp makes use of the C++ Standard Graphics proposal which you can find at github.com/cpp-io2d.

Thanks to the io2d team for keeping things going.

Thanks also to Hana Dusíková for prompting me to create a piece of open source C++ presentation software. Your move...